

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

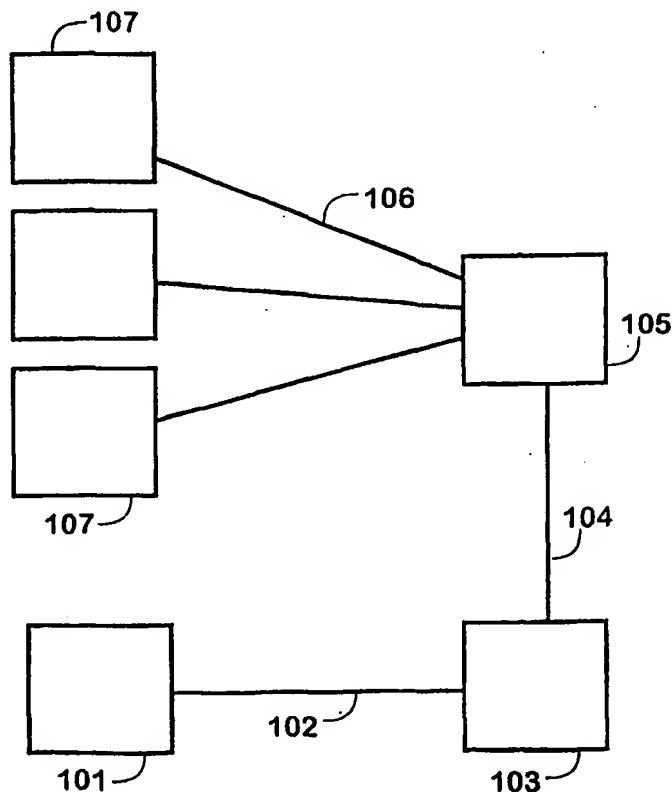
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 3/14, 15/62, G06T 1/00	A1	(11) International Publication Number: WO 00/20960 (43) International Publication Date: 13 April 2000 (13.04.00)
(21) International Application Number: PCT/US99/23051 (22) International Filing Date: 4 October 1999 (04.10.99) (30) Priority Data: 60/103,042 5 October 1998 (05.10.98) US (71)(72) Applicant and Inventor: KEEHAN, Michael, T. [US/US]; 1277 Los Arcos Drive, Prescott, AZ 86305 (US). (74) Agents: CANNIFF, Brian, P. et al.; Elman & Associates, 20 West Third Street, P.O. Box 1969, Media, PA 19063-8969 (US).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: **ASYNCHRONOUS VIDEO FORUMS**

(57) Abstract

An asynchronous video forum is the maintenance of a periodic or ongoing interaction of a group on a focused topic. "Thought leaders" are provided with forum technology so that they can interact with other asynchronous video forum participants (107) on a focused group. Knowledge providers (101) are invited to record and/or upload their video clips to a receiving server (102). The clips are converted to popular and efficient "streaming" file formats, then placed in appropriate "conversational" sequence on a video-stream server (105). A video-stream server (105) is provided on a computer network (106), such as the Internet. Users (107) are allowed to connect to the video-stream server (105) using appropriate protocols. The viewer/participant sees a sequence of "talking head" shots of participant comments that build on previous comments. The major difference in viewing asynchronous video forum "meeting" is that the comments would have been input over many hours or even days by people who might be widely dispersed geographically.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

ASYNCHRONOUS VIDEO FORUMS

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the
5 facsimile reproduction by anyone of the patent disclosure, as it appears in the PTO patent file or records, but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

The present invention generally relates to information systems and particularly to systems facilitating an electronically-mediated, asynchronous
10 discussion among knowledge providers, distributing information and entertainment to consumers.

SUMMARY OF THE INVENTION

There are new communications capabilities which will naturally emerge as a result of technological trends which are easily identifiable today. For example, with
15 the increased availability of broadband computer communications capability and reductions in the cost of computer data storage, it seems inevitable that "video mail" will become an important alternative and adjunct to electronic mail.

Video mail differs from video conferencing in that the message is recorded by the sender, stored on a computer disc and forwarded to the receiver on his or her
20 command. This difference is analogous to the difference between computer "chat" systems and electronic mail; with mail, the sender and receiver need not be accessing the computer system at the same time. Electronic mail and video mail are "store-and-forward" media, allowing messages to be sent and received at the convenience of the user.

Neither video conferencing nor chat systems have enjoyed the degree of
25 acceptance which was anticipated. There are various reasons why this is the case, but it is clear that the addition of store-and-forward capacity to a chat system results in a communications system of well established value, i.e. electronic mail. Similarly, the addition of "store and forward" capability to video conferencing produces a
30 communications system that will likely have much broader application and acceptance than real-time video conferencing has enjoyed to date.

This store-and-forward version of video conferencing is called video mail. As communications bandwidth increases and storage cost decreases, it becomes more viable. The development and emergence of video mail leads to efforts to use the new capability to facilitate communications among groups of people on a focused topic. That is, what might be called "video forums" will naturally develop from the occasional need to convene a group discussion on a focused topic using video mail technology.

The applications of video forums, while important, are anticipated by the present invention. Clearly, if a group can meet face-to-face to deal with a group communications need, they should do so. If computer conferencing or a bulletin board system would meet their needs, they should look no further in their communications toolbox. However, given certain subject matter, conditions and requirements, an asynchronous video forum will suggest itself as a valuable capability. The present invention provides an "asynchronous video forum."

An asynchronous video forum is the maintenance of a periodic or ongoing interaction of a group on a focused topic. "Thought leaders" are provided with video forum technology so that they can interact with other asynchronous video forum participants on a focused topic.

The analogy of a moderated face-to-face roundtable discussion is useful in understanding an asynchronous video forum. The moderator introduces the topic, provides background, and then calls on the first group member for comment. A second member of the group is asked to respond, followed by a third and so on around the table. Once everyone has had a turn to speak (or pass), the sequence begins again after the moderator has had a chance to summarize or make some process comment.

The experience of watching the video record of such a meeting is similar to watching the recording of a round-the-table discussion where each participant has her own video camera sitting before her on the conference table. The viewer sees a sequence of "talking head" shots of participant comments that build on previous comments. The major difference in viewing an asynchronous video forum "meeting" is that the comments would have been input over many hours or even days by people who might be widely dispersed geographically.

The experience from the standpoint of any one of the participants is very different from a roundtable discussion, although familiar in terms of process. That is, participants sit down at computers appropriately equipped with video recording and playback capability. They "sign-on" the system which tells them that they have been invited to take part in a certain asynchronous video forum. On command, the system plays for them all of the comments that have been recorded by participants to date. For example, the first clip might be a 3 to 5 minute opening statement by the moderator establishing the topic, setting ground rules for the discussion (e.g. brevity, order of comments, etc.), and asking participants to introduce themselves. A participant reviews the introductions of those who had submitted them to date and submits his or her own. Each participant has theoretically unlimited time to prepare her submission. However, the next person to sign on after the submission is made sees that submission as though there were no break in the continuity of the discussion. That is, each participant signing on has the illusion that she was seeing the record of a discussion that was recorded in "real time". In fact, since any asynchronous video forum participant would have "unlimited" time to record submissions, she is able to select from several alternative "takes". Under appropriate circumstances, a participant is able to edit into their submission an outline of discussion points, graphics or even location video if it seemed useful in making a point.

The importance of the moderator's role cannot be understated in this application. Process comments are essential from time-to-time in order to keep the discussion on track. Clearly there is a learning curve for participants. Some participants will "take" to this medium more readily than others; some will be judged as having more "talent" than others. Something like a screen-test would be required if the video record of an asynchronous video forum is to hold the attention of anyone other than the participants.

It is intended that the video record of an asynchronous video forum will ultimately be of interest to a larger audience than is available currently on the Internet. It is seen as a new medium for adult education. Furthermore, it is hoped that such educational products will have entertainment value as is the case with public television, the learning channel, the history channel, etc. Indeed, it is hoped that the resulting "programs" might at some point be of sufficient interest and value

that they might be not only made available through the internet, but also be broadcast to a larger audience on television. One can imagine an ongoing asynchronous video forum being carried by a cable station and thought of as one might think of live discussion groups such as "Washington Week in Review" or others. Depending on the quality of the discussion and the enduring interest of the topic, archival storage and availability may prove valuable as well.

Initial applications, however, will involve simple access to the evolving record of the discussion by suitably equipped "view only" subscribers over the Internet. That is, when a viewer who has access to view the discussion signs on the system, they would be able to view all of the video "comments" made by participants since that viewer last signed on. In group process terminology, this is analogous to a "fishbowl" format. It is also analogous to "read only" participation in a computer conferencing system. In fact, there could be a parallel computer conferencing discussion on the topic of the fishbowl with submission of questions or comments to the asynchronous video forum through the moderator.

It is an object of the present invention to provide an electronically-mediated discussion among knowledge providers for the purpose of distributing information and entertainment to consumers.

-Knowledge providers are invited to record and/or upload their video clips to a receiving server. The clips are converted to popular and efficient "streaming" file formats, then placed in appropriate "conversational" sequence on a video-stream server.

A video-stream server is provided on a computer network, such as the Internet. Users are allowed to connect to the video-stream server using appropriate protocols. For example, a user can click on an appropriate link or URL on a webpage and be directed to the server where the information resides. As another example, the server may be part of a company's Intranet. Once connected to the video-stream server, a program (e.g. CGI script) on the server transmits parameters that identify where (e.g. which host/which URL) the user came from, and what she wants to see. If a browser-implemented version is used, it may determine whether not the person has seen any particular item in the collection. This determination may be made through the use of information stored on the server (e.g. in a user-

account directory or user-account database) or information kept on the visiting person's computer (e.g. through the use of cookies).

In a preferred implementation, available files are listed on a webpage. A CGI front end allows users to select one or more video clips. The CGI-based front end
5 accesses a control file containing information about all available clips, preferably including at least clip ID, date and time, and "title/subject" which may include presenter's name and topic.

The CGI-based script, perl script or other script on the server generates a datastream that is MIME-typed as .ASF. This typing identifies the datastream to the
10 user's browser and user's system software as a set of instructions to be passed to the user's viewing software (e.g. MICROSOFT MediaPlayer in the case of an .ASF file). The browser activates the user's viewing software which then contacts another port on the host that hosts the desired video clip(s). Ideally, the servers involved should have enough processing power (e.g. dual PENTIUM 400Mhz machine or
15 other machine as appropriate) and memory (e.g. 256 megabytes) for the processor- and memory-intensive job of setting up appropriate video streams for all users. The server should also have a large bandwidth connection (e.g. a dedicated T1), so that multiple videostreams can be broadcast to different users.

Access to the stored clips may be open (e.g. to anyone who can find it) or
20 restricted. Restriction optionally includes creation of accounts and passwords. Restriction optionally includes limitations based on IP address if content is not meant to be viewed in certain areas of the country, world, Intranet, network, etc.

The present invention is highly valuable in that it can generate an ongoing knowledge base created by this knowledge management technology. The records
25 of the various discussions may be researched, annotated, indexed by heading and/or speech-to-text transcripts, transcribed, etc.

The present invention provides a computer system for managing an electronically-mediated, asynchronous discussion among knowledge providers, comprising: a first connective means from a first computer to a second computer; a
30 second computer; said first connective means allowing a knowledge provider operating a first computer to capture clip information, encode clip information, and upload an appropriately formatted clip from a first computer to said second computer; a second connective means allowing said second computer to

communicate said appropriately formatted clip to a third computer; said third computer; and a third connective means allowing said third computer to communicate said appropriately formatted clip to at least one viewer.

Another embodiment of the present invention further comprises a knowledge provider serves as a discussion leader and provides additional background information to said knowledge providers through said connective means. Another embodiment of the present invention further comprises a process manager to organize the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested by individual viewers and knowledge providers.

Another embodiment of the present invention further comprises a process manager to organize and edit the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested by individual viewers and knowledge providers.

Another embodiment of the present invention further comprises said process manager comprises management software residing on said second computer. Another embodiment of the present invention further comprises a fourth connective means to allow viewers to communicate feedback to a feedback manager; said feedback comprising at least one of the group of emails, telephone calls, multimedia and text responses, survey tools, television set-top box technology input; and said feedback manager communicating said feedback to said discussion leader.

Another embodiment of the present invention further comprises summarizing means for sifting, sorting and summarizing said feedback for said feedback manager to communicate a summary of said feedback to said discussion leader.

Another embodiment of the present invention further comprises a first connective means from a first computer to a second computer; a second computer; said first connective means allowing a knowledge provider operating a first computer to capture clip information, encode clip information, and upload an appropriately formatted clip from a first computer to said second computer; a second connective means allowing said second computer to communicate said appropriately formatted clip at least one viewer; and a process manager to organize the clips into discussion form, authorize viewers and knowledge providers, track which clips have been

viewed by individual viewers and knowledge providers and transmit clips as requested by individual viewers and knowledge providers.

Another embodiment of the present invention further comprises a fourth connective means to allow viewers to communicate feedback to a feedback manager; said feedback comprising at least one of the group of emails, telephone
5 calls, multimedia and text responses, survey tools, television set-top box technology input; and said feedback manager communicating said feedback to said discussion leader.

Another embodiment of the present invention comprises providing a first
10 connective means from a first computer to a second computer; providing a second computer; providing said first connective means allowing a knowledge provider operating a first computer to capture clip information, encode clip information, and upload an appropriately formatted clip from a first computer to said second
15 computer; providing a second connective means allowing said second computer to communicate said appropriately formatted clip to a third computer; providing said third computer; and providing a third connective means allowing said third computer to communicate said appropriately formatted clip to at least one viewer.

Another embodiment of the present invention further comprises providing a knowledge provider serves as a discussion leader and provides additional
20 background information to said knowledge providers through said connective means.

Another embodiment of the present invention further comprises providing a process manager to organize the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested by individual viewers and
25 knowledge providers.

Another embodiment of the present invention further comprises providing a process manager to organize and edit the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested by individual
30 viewers and knowledge providers.

Another embodiment of the present invention further comprises a fourth connective means to allow viewers to communicate feedback to a feedback manager; said feedback comprising at least one of the group of emails, telephone

calls, multimedia and text responses, survey tools, television set-top box technology input; and n) said feedback manager communicating said feedback to said discussion leader.

Another embodiment of the present invention further comprises providing
5 summarizing means for sifting, sorting and summarizing said feedback for said feedback manager to communicate a summary of said feedback to said discussion leader.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention, both as to its organization and its manner of operation,
10 together with further objects and advantages, may be best understood by reference to the following description, in connection with the accompanying drawings.

Fig. 1 shows hardware helping to illustrate an embodiment of the present invention.

Fig. 1A shows hardware helping to illustrate an embodiment of the present
15 invention.

Fig. 2 shows a diagram showing a flow of information in the present invention.

Fig. 3 shows a diagram showing a flow of information in the present invention.

Fig. 4 shows a diagram showing a flow of information in the present invention.

DEFINITIONS

20 Asynchronous: occurring at various times.

Knowledge Provider: an individual who is knowledgeable on the topic of the forum, and who is selected to provide information in the form of clips to the forum discussion. Knowledge providers are typically geographically dispersed and few in number. In special cases, a computerized process may serve as a knowledge
25 provider.

Clip: a segment of multimedia information electronically transmitted by a knowledge provider as a comment in the discussion.

Discussion: a stream or series of clips in an asynchronous video forum which represent the record of a building discussion among knowledge providers on the
30 topic of the forum.

Discussion leader: the individual who introduces the topic of a forum and manages the ensuing discussion. A discussion leader may also be a knowledge

provider and/or the feedback manager. In special cases, the discussion leader may have editorial control over the content of the clips.

Feedback Manager: receives feedback on the forum from consumers. That feedback is then sifted, sorted, summarized and entered into the discussion through either the discussion leader or an automated process.

Feedback: information, opinions or questions provided by consumers to the forum through the feedback manager. Survey feedback and content analysis tools may be used by the manager to sharpen consumer feedback.

Participants: knowledge providers and consumers who have access to a particular discussion.

Process manager: a person and/or automated process which among other things organizes the clips into discussion form, authorizes participants, tracks which clips have been viewed by individual participants and transmits clips as requested by individual participants. In special cases, the process manager may have editorial control over the content of the clips. This control would preferably rest with a human discussion leader.

Consumer: an individual who requests access to a forum for the purpose of learning or being entertained. Any consumer may provide feedback (text or multimedia) to the asynchronous video forum through the discussion leader. The number of consumers is limited only by access to the electronic medium on which the asynchronous video forum is available.

DESCRIPTION OF PREFERRED EMBODIMENTS

Referring now to Fig. 1, the knowledge provider 101 produces a video file on her own machine, or receives such a file from someone else. If the knowledge provider 101 produces the video file, it may be done in any of a number of ways. One way is through the use of commercially available cameras (such as the CONNECTIX ball camera) and microphones (whether integrated with the camera or disconnected therefrom) to create an appropriate AVI, MOV, MPEG or other format file. Alternately, the file is already made available in a ready-to-distribute format, such as MICROSOFT's ASF (Advanced Streaming Format).

The knowledge provider 101 fills in information about the author (e.g. email, name) and subject (e.g. a header line), selects the file to upload and initiates the transfer of the video file. The knowledge provider 101 then uploads the file through

connective means 102 to the first server 103. This upload can be performed using file transfer protocol (FTP) software or done through appropriate means through a web browser. Connective means 102 is an Internet connection, phone line, network connection, being at the terminal or keyboard of the first server 103, or equivalents thereto.

In order for the video clip to be made available to as many users as possible, it preferably is made available in one or more popular video file formats. For example, the uploaded video clips are run through Microsoft's NetShow Encoder to produce ASF files. It is an advantage of ASF files that that they play as they 'stream' to the receiver in ways that make them more efficient than larger AVI or MPEG files. Other file formats, such as REALAUDIO's RealVideo format, so-called "thin server" VDO videostreaming technology, or other technologies made available on various operating systems are alternatives. Other existing and/or to-be-invented file formats which have desired compression and streaming playback capability are also useful for the present invention (e.g. VRML and/or other 3D representations).

The encoding process is preferably automated. For example, encoding takes place on receipt of file(s), or on a timed schedule through the use of software written for this purpose.

If during the encoding process there are any error indications, the original knowledge provider 101 of the file can be contacted. If there is no error, however, the encoded file is then moved from first server 103 through appropriate connective means 104 (e.g. network connection, Internet connection) to the second server 105.

Once the "final" form of a given video clip exists on the second server 105, it's ready to be "posted." Posting will include (at least) moving the clip into the correct directory, with an appropriate name. Posting will also include entering the appropriate name into a list that will be used by the download management software. The name and some information about the file is then optionally made available to potential viewers (e.g. a pull-down list). Preferably, the list refers to some information (e.g. login account, cookies set by the browser, if any) to keep track of who has already seen what clips.

Preferably, an automatic notification is periodically sent out to an appropriate user base, through the use of a listserve or other appropriate means.

Preferably, system software should provide manager the opportunity to create their own new discussion topic or "panel" through the use of a template. Although the process might be implemented on paper or otherwise require human interaction, the panel creation process may be automated through the use of forms presented in a web browser. At that point, or later in a human-governed approval process, appropriate access rights for editing clips, notification, etc. are created.

In a preferred embodiment, a first server receives files from knowledge providers. This first server then places the files in an appropriate place for encoding to take place. Encoding management software automatically detects the presence and type of files, then activates an appropriate encoding program to convert the file to a popularly available format of choice, such as ASF. The ASF version of the video file is then passed to an appropriate spot on the video-streaming server. Preferably, the video-streaming server has excellent processor power, large amounts of memory, and a large-bandwidth connection to viewers. Viewers, whether knowledge providers participating in the discussion or consumers having 'read only' access to the stored clips, visit an interface to the stored video clips which tells them who wrote the clip, a subject heading, and when the clip was recorded and published. The interface also contains some indication as to what clips the viewer has already seen through the use of cookies placed on the viewer's machine or though user account information stored elsewhere (preferably on the video-streaming machine).

In another preferred embodiment, a first server receives files from knowledge providers. This first server then places the files in an appropriate place for encoding to take place. Encoding management software is triggered by a human operator or on an appropriate schedule. The encoding management software automatically detects the presence and type of files, then activates an appropriate encoding program to convert the file to a popularly available format of choice, such as ASF. Once the file has been converted, the original file is deleted from the system. The ASF file remains on the first server until moved to the video-streaming server by appropriate software or by a human agent. The ASF version of the video file is then passed to an appropriate spot on the video-streaming server.

In a preferred embodiment, each forum has at least one discussion leader. The discussion leader is an individual who introduces the topic of a forum and

manages the ensuing discussion. A discussion leader may also be a knowledge provider and/or the feedback manager.

In a preferred embodiment, each forum is provided with a feedback manager. The Feedback Manager receives feedback on the forum from consumers. That
5 feedback is then sifted, sorted, summarized and entered into the discussion through either the discussion leader or an automated process. Feedback may be responses, information, opinions or questions provided by consumers to the asynchronous video forum through the feedback manager. The feedback manager may use survey feedback and/or content analysis tools may be used to sharpen consumer feedback.
10 This iterative process of sharpening the focus of and/or redirecting the discussion through feedback will be crucial to many scaled applications. When the volume of feedback is large, content analysis may be used on the feedback received. Alternatively, a representative sample is used to focus in on the particulars of the feedback. For this reason, and others, consumer feedback is crucial from a
15 communication standpoint.

In a less preferred embodiment, the Process Manager role is split between software and a human agent. Software is used to authorize consumers, track which clips have been viewed by individual participants, and transmits clips as requested by individual participants. A human agent manages the selection of knowledge
20 providers (e.g. approving the 'talking heads' who may participate), and in addition may interject process clips to keep the discussion on point. In special cases, the process manager may have editorial control over the content of the clips.

In a preferred embodiment, software is used to authorize consumers, track which clips have been viewed by individual participants, and transmits clips as
25 requested by individual participants. An optional human agent discussion leader manages the selection of knowledge providers (e.g. approving the 'talking heads' who may participate), and in addition may interject process clips to keep the discussion on point. In special cases, the process manager may have editorial control over the content of the clips.

30 Referring now to Fig. 1A, a knowledge provider 111 uploads a clip through connective means 112 to a single server 113. If necessary, encoding management software performs any necessary or desired conversion of file formats. The single server 114 is also used as the video-streaming server. The single server 113

provides video streams across connective means 114 to various viewers 115. This implementation is likely to be popular for asynchronous video forums conducted across an Intranet.

Referring now to Fig. 2, the discussion leader 201 is the individual who introduces the topic of a forum and manages the ensuing discussion. Knowledge providers 202 receive input from the discussion leader in the form of clips and/or supporting documentation and email so that knowledge providers 202 are able to contribute appropriately to the forum. The knowledge providers 202 and the discussion leader 201 provide video clips to the video-streaming server. Process Manager 203 is a person, likely supplemented or alternatively replaced with process management software, who organizes the clips into discussion form, authorizes participants, tracks which clips have been viewed by individual participants and transmits clips as requested by individual participants. In special cases, process manager 203 may have editorial control over the content of clips. Consumers or users 204 are individuals who request access to a forum for the purpose of learning or being entertained. Consumers 204 may provide feedback to the forum through the Discussion Leader 201. The number of consumers 204 is limited only by access to the electronic medium on which the forum is available. The Feedback manager 205 receives feedback on the forum from consumers 204. The Feedback may be in the form of emails, telephone calls, multimedia and text responses, survey tools, television set-top box technology, or other means of gathering information on the discussion. The feedback is summarized after having been sifted and sorted using tools appropriate to the scope of the feedback desired and/or received from consumers 204.

Referring now to Fig. 3, the discussion leader 301 is the individual who introduces the topic of a forum and manages the ensuing discussion. Knowledge providers 302 receive input from the discussion leader in the form of clips and/or supporting documentation and email so that knowledge providers 302 are able to contribute appropriately to the forum. The knowledge providers 302 and the discussion leader 301 provide video clips to the video-streaming server. Process Manager 303 is a person, likely supplemented or alternatively replaced with process management software, who organizes the clips into discussion form, authorizes participants, tracks which clips have been viewed by individual participants and

transmits clips as requested by individual participants. In special cases, process manager 303 may have editorial control over the content of clips. Consumers or users 304 are individuals who request access to a forum for the purpose of learning or being entertained. Consumers 304 may provide feedback to the forum through the Discussion Leader 301. The number of consumers 304 is limited only by access to the electronic medium on which the forum is available.

Referring now to Fig. 4, the discussion leader 401 is the individual who introduces the topic of a forum and manages the ensuing discussion. Knowledge providers 402 receive input from the discussion leader in the form of clips and/or supporting documentation and email so that knowledge providers 402 are able to contribute appropriately to the forum. The knowledge providers 402 and the discussion leader 401 provide video clips to the video-streaming server. Process Manager 403 is process management software which organizes the clips into discussion form, authorizes participants, tracks which clips have been viewed by individual participants and transmits clips as requested by individual participants. In special cases, discussion leader 403 may have editorial control over the content of clips. Consumers or users 404 are individuals who request access to a forum for the purpose of learning or being entertained. Consumers 404 may provide feedback to the forum through the Discussion Leader 401. The number of consumers 404 is limited only by access to the electronic medium on which the forum is available.

Scripts

The following scripts for use on a website illustrate many of the features of the present invention discussed herein.

getvid.pl - Script determines from the query string with which it is invoked
5 (e.g. `http://host.net/getvid.pl?salon`) for which asynchronous video forum a file is to be received. It determines the local name it will use for the file as a character string which is the current date and time as a hex string. It produces an HTML form with the Asynchronous video forum identifier and the file identifier as hidden input fields, and fields for the user the provider's name, the title or other identifying
10 information on the video clip, and the name of the file on the sender's system to be transferred. The web browser collects the information and transmits it, together with the actual file, to the `getvid.asp` script.

getvid.asp - Receives from the user's browser the contents of the form generated by `getvid.pl`, and the file transmitted from the user's computer. The file
15 is copied into a receiving directory appropriate to the particular asynchronous video forum, with the local name made up by the previous script. Another file, with the same name but the extension `.FRM`, is also created in the same directory, containing the other information collected by the previous form. These files in turn are used by the `encode.pl` script.

encode.pl - Is run from time to time by an administrator. The program checks
20 and sets interlocks to make sure that it is not being run multiple times simultaneously. It iterates through the directories belonging to the various asynchronous video forums, looking for `.FRM` files. It iterates through those files, and for each one it finds, it invokes the MICROSOFT NetShow Encoder for the
25 corresponding video file. (If the corresponding file is of type `.ASF`, meaning it was encoded before being transmitted, then the encoding step is skipped and the file is simply moved to where it has to go.) The resulting `.ASF` file is placed in a directory appropriate to the Asynchronous video forum. A file called `DATA.TXT` in that same directory is updated with information from the original form, giving the
30 provider's name and clip information, and the local name (the hex string) of the file. This data is used by `display.pl`

display.pl - Tests to see if an HTTP "cookie" has come from the user's web browser indicating which clip in the Asynchronous video forum's series she last saw. Produces a welcome page to the Asynchronous video forum, which can be a coded-in default page or can be an alternative made available as the file
5 BASE.HTML in the Asynchronous video forum's content directory. Within this page (either way it's generated) the script inserts a form allowing the user to pick which clips she wishes to view. If the cookie was available, and if new clips have been added since the last one viewed, those new clips are automatically pre-selected, though the user can change this at will. Information identifying the
10 cookie value and the identity of the Asynchronous video forum are included in the form as "hidden" fields. This information is used by display.pl1

display1.pl - Produces an output stream identified with the MIME type application/x-ms-asf, which is a text stream conforming to Microsoft's specification for a Version 3 Advanced Streaming eXtension (or .ASX) file. This MIME type
15 tells the user's browser (which has been set up with this information) to invoke the MS Media Player as the application to handle this information. If the user did not previously have a cookie value set (detected by the previous script) or if the last clip viewed in this run is a later one than the previous last viewed, the user's browser is requested to set a new cookie value for subsequent use.

20

DISPLAY.PL

```

#!perl
~
5  # This program uses either a provided HTML file or its own default data
  # file to produce the basic structure and most of the content of an HTML
  # page laying out the list of clips available in a particular salon. If
  # the user has visited this salon before, and allowed a cookie to be set,
  # then those clips that came later than the last previously selected will
10 # be selected automatically.
  #
  # If an HTML file is provided, it must be placed in the salon's content
  # directory with the name BASE.HTML. It must include the token "__BREAK__"
  # on a line by itself at three locations. The first two will each be
15 # "filled" with the name of the salon, and are designed to be within
  # the <TITLE></TITLE> tags and within the body. Either or both can be
  # commented out in the file
  # <!--
  # __BREAK__
20 # -->
  # if it is desired not to use that information. The third __BREAK__
  # will be replaced by a form which will invoke the program that actually
  # results in video clips being streamed to the user. If it is not used,
  # then there's no reason to use this package!
25 use Html;
  package main;
  $formAction = 'http://www.KnowledgeAssembly.net/Scripts/display1.pl';
  $rootDir = 'c:\inetpub\vidfiles';
  $query = $ENV{QUERY_STRING} or (noQuery("none"),exit 0);
30 if ($query eq "") {noQuery("none");}
  elsif ($query eq 'test') {
    $salon = 'Test';
    $dirName = 'test';
    $mainDir = "$rootDir\\$dirName\\content";
35    doSalon();
  }
  elsif ($query eq 'mb') {
    $salon = 'Bill and Michael';
    $dirName = 'test';
40    $mainDir = "$rootDir\\$dirName\\content";
    doSalon();
  }
  #__ADD_HERE
  else {
45    noQuery("bad");
  }
  exit 0;
  #
50 sub cantCD {

```

```

print "Content-type: text/plain\n\nCan't CD to $mainDir.\n";
exit 0;
}

5  sub doSalon {
    cantCD() unless chdir($mainDir);
    my @data;
    my $lastSeen;
    $htmlFile = 'BASE.HTML';
10  if (open( HTMLFILE, "<$htmlFile" )) { $htmlFile = \'HTMLFILE; }
    else { $htmlFile = \'DATA; }
    { #to reclaim memory from following "my" variable(s)
      my @tempData;
      problem('Can\'t open data.txt file.')
15      unless open DATAFILE, "<data.txt";
      @tempData = <DATAFILE>;
      close DATAFILE;
      foreach $line (@tempData) {
        { # so each new "array" is new
20        my @array = split /\|/, $line;
        push @data, \@array;
        }
      }
      # @data is now an array of references to arrays of the data, which
25      # are filename, presenter, and description.
      my $cookies = Html->getCookies();
      $lastSeen = $$cookies{$dirName.'_last'} || '8FFFFFFF';
    }
    print "Cache-Control: no-cache\n";
30    print "Pragma: no-cache\n";
    print "Content-type: text/html\n\n";
    tilBreak();
    print "\n$salon\n"; # title
    tilBreak();
35    print "\n$salon\n"; # in body
    tilBreak();
    print qq(<FORM ACTION="$formAction" METHOD="GET">\n);
    print qq(<INPUT TYPE="HIDDEN" NAME="directory" VALUE="$dirName">\n);
    print qq(<INPUT TYPE="HIDDEN" NAME="salon" VALUE="$salon">\n);
40    if ($lastSeen ne '8FFFFFFF') {
      print qq(<INPUT TYPE="HIDDEN" NAME="lastSeen" VALUE="$lastSeen">\n);
    }
    print qq(
      <B>The speed of my Internet connection is:</B><BR>
45      <INPUT TYPE="RADIO" NAME="speed" VALUE="s"> Slower than 33.6Kbps
      <INPUT TYPE="RADIO" NAME="speed" VALUE="m" CHECKED> 33.6 thru
56Kbps
      <INPUT TYPE="RADIO" NAME="speed" VALUE="f"> Faster than 56Kbps<P>
    );
50    my $newSwitch = 0;

```

```

my $oldSwitch = 0;
# Two select fields in form. First is "new" clips...
my $lineCount = @data;
$lineCount = 9 if $lineCount>9;
5  foreach $entry (@data) {
    if (($lastSeen lt $$entry[0])) {
        if (!$newSwitch) {
            print "New clips...<BR>\n";
            print qq(<SELECT NAME="newSelection" SIZE=$lineCount MULTIPLE>\n);
10         $newSwitch = 1;
        }
        print qq(<OPTION VALUE="$$entry[0]|$$entry[1]|$$entry[2]");
        print " SELECTED>$$entry[1]: $$entry[2]\n";
    }
15 }
    if ($newSwitch) {
        print qq(</SELECT>\n);
    }
    # ...then "old" clips...
20  foreach $entry (@data) {
    if (($lastSeen lt $$entry[0])) {
        if (!$oldSwitch) {
            if ($newSwitch) {print "<BR>";}
            print "Previously viewed clips...<BR>\n";
25         print qq(<SELECT NAME="oldSelection" SIZE=$lineCount MULTIPLE>);
            $oldSwitch = 1;
        }
        print qq(<OPTION VALUE="$$entry[0]|$$entry[1]|$$entry[2]");
        print ">$$entry[1]: $$entry[2]\n";
30     }
    }
    if ($oldSwitch) {print qq(</SELECT>\n);}
    if ($newSwitch && $oldSwitch) {
        print qq(<P>
35     <INPUT TYPE="submit" NAME="viewOld" VALUE="View marked old clips
only"><BR>
        <INPUT TYPE="SUBMIT" NAME="viewAll" VALUE="View new AND old
selected clips"><BR>
        <INPUT TYPE="RESET" VALUE="Reset selections">
40     );
    }
    elsif ($newSwitch) {
        print qq(<P>
        <INPUT TYPE="SUBMIT" NAME="viewAll" VALUE="View selected clips"><BR>
45     <INPUT TYPE="RESET" VALUE="Reset selections">
        );
    }
    elsif ($oldSwitch) {
        print qq(<P>

```

```

    <INPUT TYPE="SUBMIT" NAME="viewOld" VALUE="View selected
clips"><BR>
    <INPUT TYPE="RESET" VALUE="Reset selections">
    );
5    }
    print "</FORM>\n";
    tilBreak();
}

10 sub problem {
    my $cause = shift;
    print "Content-type: text/plain\n\n$cause\n";
    exit 0;
}

15 sub noQuery {
    print
    'Content-type: text/html

20 <HTML><HEAD><TITLE>Error notice</TITLE><HEAD>
    <BODY>
    <B>
    <FONT SIZE="+2">
    Error notice:
25 </FONT>
    <FONT SIZE="+1">
    This script has been invoked incorrectly. If you arrived here via a link
    from another web page, please advise the owner of that page.
    </FONT>
30 </B>
    </BODY></HTML>';
    }

    sub tilBreak {
35 my $line;
        while ($line = <$htmlFile>) {
            return 1 if $line eq "__BREAK__\n";
            print $line;
        }
40 return 0; # meaning we hit "end of file"
    }
    __END__
    <HTML>
    <HEAD>
45 <TITLE>
    __BREAK__
    CyberSalon
    </TITLE>
    </HEAD>

```

<BODY BACKGROUND="" BGCOLOR="#0000ff" TEXT="#ffffff" LINK="#ff0000"
VLINK="#ff80c0" ALINK="#00ff00">

Welcome to

<center>

5 <H2>

The

__BREAK__

CyberSalon

</H2>

10 </center>

Please pick the clips you would like to view from the list below:

__BREAK__

</BODY>

</HTML>

15 •

DISPLAY1.PL

```
#!/perl
~
5 use Html;
~
package main;
~
10 ~
    Html->getFormData( \%formData );
    ~
    $directory = $formData{directory};
    $oldSelection = $formData{oldSelection};
    15 $newSelection = $formData{newSelection};
    $host = "mms://www.knowledgeassembly.net";
    $homeDir = "C:\\inetpub\\vidfiles";
    $salon = $formData{salon};
    $lastSeen = $formData{lastSeen};
    20 if (defined($oldSelection)) {
        if (ref($oldSelection) eq 'ARRAY') { #user made multiple selections
            #@oldSelection = sort @oldSelection;
        }
        else { #turn it into an array, for later processing
    25     my @array;
        push @array, $oldSelection;
        $oldSelection = \@array;
        }
    }
    30 if (defined($newSelection)) {
        if (ref($newSelection) eq 'ARRAY') { #user made multiple selections
            #@newSelection = sort @newSelection;
        }
        else { #turn it into an array, for later processing
    35     my @array;
        push @array, $newSelection;
        $newSelection = \@array;
        }
    }
    40 { #temporary variable scope
        my @array;
        push @array, @oldSelection if @oldSelection;
        undef $oldSelection;
        45 if ($formData{viewAll}) { push @array, @newSelection if @newSelection; }
        undef $newSelection;
        $selection = \@array;
    }
    if (! @selection) {
    50     print "Content-type: text/html\\n\\n";
    }
```



```

print "<HTML><HEAD><TITLE>\n";
print "ERROR\n";
print "</TITLE></HEAD><BODY>\n";
print qq(<FONT size="+2">\n);
5 print qq(<B>Error:</b>);
print qq(Please press your "Back" button and select at least one clip );
print qq(to view. Thank you.);
print qq(</FONT></BODY></HTML>\n);
exit 0;
10 }
@$selection = sort @$selection;
#Now, $selection is a reference to a sorted array, of one or more members.
$lastSelection = $$selection[ @$selection - 1];
$lastSelection =~ m/!/;
15 $lastSelection = $`; #now contains the key value for the latest clip selected
if (($lastSeen && ($lastSelection gt $lastSeen)) || !$lastSeen) {
    Html->setCookie($directory."_last=$lastSelection", 30);
}
print "Content-type: video/x-ms-asf\n\n";
20 {
    my @tmp = localtime();
    $year = 1900+$tmp[5];
}
print qq(<ASX Version="3">\n\n<COPYRIGHT>Knowledge Assembly Inc.
25 $year</COPYRIGHT>\n);
print qq(\t<title>$salon</title>\n);
foreach $line (@$selection) {
    my @data = split /\|/, $line;
    print "\t<ENTRY>\n";
30 my $speed = $formData{speed} || 'm';
CHECK_SPEED:
if (! -f("$homeDir\\$directory\\content\\$data[0]$speed.asf")) {
    # No file, or no file for that speed
    if ($speed eq 'f') {$speed = 'm'; goto CHECK_SPEED;}
35 if ($speed eq 'm') {$speed = 's'; goto CHECK_SPEED;}
    if ($speed eq 's') {$speed = ""; goto CHECK_SPEED;}
    if ($speed eq "") {
        # Got a real problem. No file.
        goto SKIP_ENTRY;
40 }
}
print
    qq(\t\t<ref href="$host/$directory/$data[0]$speed.asf" />\n);
print qq(\t\t<author>$data[1]</author>\n);
45 {
    my $substr = substr($data[2],-1);
    if ("n" eq substr($data[2],-1)) {chop $data[2]; chop $data[2];}
}
print qq(\t\t<title>$data[2]</title>\n);
50 print qq(\t</ENTRY>\n);

```

SKIP_ENTRY:

}

print qq(</ASX>\n);

#

5

•

ENCODE.PL

```

package main;

5
~
# $vidToASF = 'vidtoasf';
~

$mainDir = 'c:\inetpub\vidfiles';
10 print "Cache-Control: no-cache\n";
print "Pragma: no-cache\n";
print ("Content-type: text/plain\n\nEncoding manager V2.c\n\n");
(print ("***Can't change directory to $mainDir.\n"),exit 0)
  unless (chdir($mainDir));

15
# prevent multiple copies of encode from running at same time...
BEGIN { $main::fileLockActive = 0; }
END {
  if ($fileLockActive) {
20     unlink $fileLockActive or die "$fileLockActive: $!";
  }
}
{
  my $lockFile = "encode.lock";
25  if (-f $lockFile) {
    print ("Another copy of $0 may be running.\n");
    print ("Lockfile '$mainDir\\$lockFile' is present. Unable to continue.\n");
    exit 0;
  }
30  else {
    $fileLockActive = open( LOCK, ">$lockFile" ) or die "$lockFile: $!";
    close LOCK;
    $fileLockActive = "$mainDir\\$lockFile";
  }
35 }

@dirList = ();
{
  my $dirlistString = `dir /b/ad/on`;
40  @dirList = split /\n/s, $dirlistString;
}

foreach $dir (@dirList) {
  print "Salon: $dir\n";
  if (chdir( $mainDir.\$dir.\upload)) {
45    my $tempListString = `dir /b/a-d/on *.frm`;
    my @tempList = split /\n/s, $tempListString;
    foreach $frmFile (@tempList) {
      print "\nProcessing $frmFile\n";
      $frmFile =~ m/./;
50    my $baseFile = $`;

```

```

my @tmpList = glob( "$baseFile.*" );
foreach $file2 ( @tmpList) {
    my ($base,$extension) = split /\./, $file2;
    $extension = uc $extension;
5   if ($extension eq 'FRM') {} # not processing
    elsif ($extension eq 'AVI' || $extension eq 'MOV') {
        #my $result = ` $vidToASF -in $base.$extension -out ../content\\$base.asf`;
        my $encoderDir = 'C:\\ecc\\';
        $ENV{'ASF_OUT'} = "$mainDir\\$dir\\content\\$base.asf";
10        $ENV{'VIDEO_IN'} = "$mainDir\\$dir\\upload\\$base.$extension";
        $ENV{'ENCODE_ASD'} = $encoderDir.'scaled.asd';
        if (! -f($encoderDir.'slides.asd')) {
            print "$encoderDir\\slides.asd doesn't exist. ";
            print "Can't continue.\n";
15            exit 0;
        }
        if (! $ENV{ENCODE_ASD}) { #not enough environment space
            print "Can't set environment variable ENCODE_ASD\n";
            print "to '${encoderDir}scaleable.asd'.\n";
20            print "possibly not enough environment space.\n";
            print "\nUnable to continue.\n";
            exit 0;
        }
        my $cmd = qq(${encoderDir}ctrlr.exe);
25        print "Command was: $cmd\n";
        if (! -f($cmd)) {
            print "Program file doesn't exist!\n";
            exit 0;
        }
30        system $cmd;
        if (! -f("$mainDir\\$dir\\content\\$base.asf")) {
            print "****Translation failed.\n\n";
            my $error = `type $base.FRM`;
            print "Form file is:\n", $error, "\n\n";
35            exit 0; #DEBUG
            unlink "$base.$extension";
            unlink "$base.FRM";
        }
        else {
40            print "Translation succeeded.\n\n";
            handleVidFile( $base, $extension );
        }
    }
    elsif ($extension eq 'ASF') {
45        `copy $base.$extension ../content\\$base.asf`;
        print "File moved to correct place.\n";
        handleVidFile( $base, $extension );
    }
    else {
50        print "****Unsupported file type '$extension'.\n";

```

```

        print "****Removing file $base.$extension and $base.FRM.\n";
        unlink "$base.$extension";
        unlink "$base.FRM";
    }
5      }
    }
  }
}

10  sub handleVidFile {
    my $base = shift;
    my $extension = shift;
    my @formData;
    if (open FORMFILE, "<$base.FRM") {
15      @formData = <FORMFILE>;
      close FORMFILE;
      print "Data from $base.FRM:\n";
      my %formHash;
      foreach $line (@formData) {
20        chop $line;
        $line =~ s/\t /g; # replace tabs with single spaces
        $line =~ s/ +/ /g; # replace multi spaces w single
        $line =~ s/\/:/g; # we use / as a field separator
        $line =~ m/=/; # split line on '='...
25      print "$`=$'\n";
        $formHash{$`} = $'; #... to put into a hash
      }
      if (enterData( $dir, \%formHash )) {
30        unlink "$base.$extension";
        unlink "$base.FRM";
        print "Files removed.\n";
      }
    }
  }
35

sub enterData {
    my $dir = shift;
    my $formHash = shift;
40    (print ("\n***Unable to open data file.\n"), return 0) unless
      (open DATAFILE, ">>$mainDir\\$dir\\content\\data.txt");
    my $outLine = "$$formHash{fileName}!$$formHash{Presenter}!".
      "$$formHash{Summary}\n";
    (print ("\n***Unable to print to data file.\n"), return 0) unless
45    (print DATAFILE $outLine);
    print ">$outLine";
    close DATAFILE;
    return 1;
  }
50

```

GETVID.PL

```

~perl
~
5  package main;

~
$where = $ENV{'QUERY_STRING'} || "";
~
if ($where eq "") {noParameter();}
10 ~
elseif ($where eq 'test') {$salon='Test';} # we'll leave that name for now
~
elseif ($where eq 'mb') { $where = 'test'; $salon='Bill and Michael';}
else {badParameter();}
15 $timeFilename = time();
print "Content-type: text/html\n";
print "Cache-Control: no-cache\n";
print "Pragma: no-cache\n\n";
# copy out all the data down to the __BREAK__ token...
20 untilBreak();
print
"This form will manage the upload of your video file for the '$salon' CyberSalon.\n";
printf( "(The file will be stored as '%08X'.)", $timeFilename );
untilBreak();
25 printf qq(<INPUT TYPE="hidden" NAME="fileName" VALUE="%08X">\n),
$timeFilename;
print qq(<INPUT TYPE="hidden" NAME="directory" VALUE="$where">\n);
untilBreak();
exit 0;

30 sub noParameter {
    print "Content-type: text/plain\n\n";
    print "Cannot continue. Identification information not provided.\n";
    exit 0;
35 }

sub badParameter {
    print "Content-type: text/plain\n\n";
    print "Cannot continue. No valid identification.";
40    exit 0;
}

sub untilBreak {
    my $line;
45    while ($line = <DATA>) {
        return 1 if ($line eq "__BREAK__\n"); # meaning we've hit a __BREAK__
        print $line;
    }
    return 0; # meaning we've hit end of data
50 }

```

```

__END__
<HTML>
<HEAD>
5  <TITLE>File upload form, V2.2</TITLE>
    </HEAD>
    <BODY BACKGROUND="" BGCOLOR="#00ffff" TEXT="#000000" LINK="#ff0000"
      VLINK="#0000ff" ALINK="#8000ff">
    <CENTER><B>
10  <FONT SIZE="+2">
      Video File Upload<P>
    </FONT>
    __BREAK__
    </B></CENTER><P>
15  <FORM ACTION="http://www.knowledgeassembly.net/Scripts/getvid.asp"
      METHOD="POST" ENCTYPE="multipart/form-data">
    __BREAK__
      Presenter's name:<BR>
      <INPUT TYPE="text" NAME="Presenter" SIZE="50" MAXLENGTH="80"
20  VALUE=""><BR>
      Clip description:<BR>
      <INPUT TYPE="text" NAME="Summary" SIZE="50" MAXLENGTH="100"
      VALUE=""><P>
      Video file name:<BR>
25  <INPUT TYPE="file" NAME="inputFile" SIZE="50" MAXLENGTH="100"
      ACCEPT="video/*"><BR>
      <INPUT TYPE="submit" NAME="submit" VALUE="Send file"> <INPUT
      TYPE="Reset">
    </FORM>
30  <P>
      <FONT COLOR="#FF0000">
      <B>Notice:</B> If file upload is interrupted by an error, please just press the
      "Send file" button again. <P>
      <B>Warning:</B> When you start the file transfer, it can take a long time to
35  complete. This will depend on the size of the file, and on the speed of your
      connection. Do <B><I>not</I></B> shut down your browser before getting a
      completion message or an error message, or the time will have been wasted.<P>
    </FONT>
      Thank you!
40  </BODY></HTML>

```

HTML.PM

```
package Html;
5  ~
  $title = "";
  ~
  $header = 0;
10 ~
  $headerSent = 0;
  ~
  ~
15 sub startPage {
  ~
    shift; # package name, not used
    ~
    $title = shift;
20 ~
    $header = "Content-type: text/html\n\n<HTML><HEAD><TITLE>\n" .
      "$title</TITLE></HEAD><BODY>\n";
    return 1;
  }
25
sub send {
  my $this = shift; # package name not used, but may be later;
  if (!$headerSent) {
    if (!$header) { $this->start(""); }
30    print $header;
    $headerSent = 1;
    $header = 0;
  }
  print shift();
35  return 1;
}

sub endPage {
  my $this = shift;
40  $this->send("\n</BODY></HTML>\n");
  $headerSent = 0;
  return 1;
}

45 sub errorPage {
  my $this = shift;
  my $errno = shift;
  my $msg = shift;
  if ($headerSent) {
50    print "\n<P><H2>Error $errno:</H2><B>$msg</B><P>\n";
```



```

    }
    else {
        $header = 0;
        print "Content-type: text/html\n\n<HTML><HEAD>",
5         "<TITLE>Error $errno</TITLE></HEAD><BODY>\n",
        "<H2>Error $errno:</H2><B>$msg</B></BODY></HTML>\n";
    }
    return 1;
}

10
sub getFormData {
    my $this = shift; #in case of future use in a class
    my($hashRef) = shift;
    my($buffer) = "";

15
    if ($ENV{'REQUEST_METHOD'} eq 'GET') {
        $buffer = $ENV{'QUERY_STRING'};
    }
    elsif (defined($ENV{'REQUEST_METHOD'})) {
20
        read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
    }
    else {
        my $line;
        print "Go:\n";
25
        while ($line = <STDIN>) {
            chop $line;
            $buffer .= $line . '&';
        }
    }

30
    foreach (split(/&/, $buffer)) {
        my($key, $value) = split(/=/, $_);
        $key = decodeURL($key);
        $value = decodeURL($value);

35
        $value =~ s/<[ \t]{0,}P[ \t]{0,}>/<P>/gi;
        $value =~ s/<P>\s*+</P>/<P>/gi; # compress multiple <P> tags.
        $value =~ s/</&lt;/g; # turn off all HTML tags.
        $value =~ s/>/&gt;/g;
40
        if ($::tagsOn) {
            $value =~ s/&lt;b>/<b>/gi; # turn on the bold tag.
            $value =~ s/&lt;/b>/</b>/gi;
            $value =~ s/&lt;i>/<i>/gi; # turn on the italic tag.
            $value =~ s/&lt;/i>/</i>/gi;
45
            $value =~ s!\cM!g; # Remove unneeded carriage returns.
            $value =~ s!\n\n!<P>!g; # Convert 2 newlines into paragraph.
            $value =~ s!\n! g; # Convert newline into spaces.
        }
        if ($$hashRef{$key}) { # then it already exists
50
            my ($ref, @array);

```

```

    $ref = $$hashRef{$key};
    if (ref($ref) eq 'ARRAY') { #then it's already a multiple
        #so do nothing
    }
5    else {
        push @array, $$hashRef{$key};
        $$hashRef{$key} = \@array; # convert existing to an array
        $ref = $$hashRef{$key};
    }
10    push @$ref, $value;
}
else { $$hashRef{$key} = $value; } # it's a simple, single value
}

15 sub decodeURL {
    $_ = shift;
    tr/+// /;
    s/%(..)/pack('c', hex($1))/eg;
20    return($_);
}

# Returns a reference to hash of cookie names and values.
sub get_cookies {
25    shift; # for possible later use as a class
    my %reply = ();
    my @cookieList = split ( /; /, $ENV{HTTP_COOKIE} );
    foreach $item (@cookieList) { #won't happen if there are no cookies.
        my ($key,$value) = split( /=|#/ , $item );
30        $reply{$key} = $value;
    }
    return \%reply; #which will be () if there were no cookies.
}

35 sub cookieDate {
    shift; # class name
    my $daysTilExpiration = shift;
    my @time = gmtime(time()+($daysTilExpiration*24*3600));
    my @days = (Sun,Mon,Tue,Wed,Thu,Fri,Sat);
40    my @months = (Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec);
    my $reply = sprintf ("%days$time[6]], %02d-$months$time[4]]-%d".
        "%02d:%02d:%02d GMT",
        $time[3], 1900+$time[5], $time[2], $time[1], $time[0]);
    return $reply;
45 }

sub setCookie {
    my $class = shift;
    my $string = shift;
50    my $days = shift;

```

```
print "Set-cookie: $string; path=/; expires=".  
    $class->cookieDate($days)."\n";  
return 1;  
}
```

5

```
1;  
package main;  
•
```

GETVID.ASP

```

<%
5      Server.ScriptTimeout = 32000

      ' use the scripting object to save the form inputs
      Set PostObject = Server.CreateObject("ActiveFile.Post")
      Set DirectoryObject = Server.CreateObject("ActiveFile.Directory")
10     DirectoryObject.Path = "C:\InetPub\vidFiles"
      Set PostObject.Directory = DirectoryObject
      PathName = "C:\InetPub\vidFiles\" &
PostObject.FormInputs("directory").Value
      FileName = PathName & "\upload\" &
15     PostObject.FormInputs("fileName").Value
      Set FileObject = CreateObject("Scripting.FileSystemObject")
      Set FileHandle = FileObject.CreateTextFile(FileName & ".FRM", TRUE)
      For Each InputItem In PostObject.FormInputs
          FileHandle.WriteLine(InputItem.Name & "=" & InputItem.Value)
20     Next
      FileHandle.Close

      ' save the data
      OrgName = Trim(PostObject.FormInputs("inputFile").Value)
25     Set FileData = PostObject.FormInputs("inputFile").File
      SaveName = FileName & ".$$$"
      OrgLen = Len(OrgName)
      For i = 0 to OrgLen-1
          If Mid(OrgName,OrgLen-i,1)=". " Then
30             FileExtension = Right(OrgName,i)
             If FileExtension<>"FRM" Then
                 SaveName = FileName & "." & FileExtension
             End If
             Exit For
35     End If
      Next
      FileData.Copy(SaveName)
      FileData.Delete

40     %>
      <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

      <HTML>
      <HEAD>
45         <TITLE>Upload Complete</TITLE>
      </HEAD>

      <BODY>
      </CENTER>
50     <B>

```

Received '<%= PostObject.FormInputs("fileName").Value %>'
file for "<%= PostObject.FormInputs("directory").Value %>" salon.
<P>

Upload complete.

5
</CENTER>
</BODY>
</HTML>
•

What is claimed is

1. A computer system for managing an electronically-mediated, asynchronous discussion among knowledge providers, comprising:

- 5 a) a first connective means from a first computer to a second computer;
- b) a second computer;
- c) said first connective means allowing a knowledge provider operating a first computer to capture clip information, encode clip information, and upload an appropriately formatted clip from a first computer to said second computer;
- 10 d) a second connective means allowing said second computer to communicate said appropriately formatted clip to a third computer;
- e) said third computer; and
- f) a third connective means allowing said third computer to communicate said appropriately formatted clip to at least one viewer.

15

2. A computer system as in claim 1, wherein:

- g) a knowledge provider serves as a discussion leader and provides additional background information to said knowledge providers through said connective means.

20

3. A computer system as in claim 2, further comprising:

- h) a process manager to organize the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested by individual
- 25 viewers and knowledge providers.

4. A computer system as in claim 2, further comprising:

- h) a process manager to organize and edit the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by
- 30 individual viewers and knowledge providers and transmit clips as requested by individual viewers and knowledge providers.

5. A computer system as in claim 3, wherein:

i) said process manager comprises management software residing on said second computer.

6. A computer system as in claim 3, further comprising:

- 5 i) a fourth connective means to allow viewers to communicate feedback to a feedback manager;
- j) said feedback comprising at least one of the group of emails, telephone calls, multimedia and text responses, survey tools, television set-top box technology input; and
- 10 k) said feedback manager communicating said feedback to said discussion leader.

7. A computer system in claim 6, further comprising:

- l) summarizing means for sifting, sorting and summarizing said feedback for
- 15 said feedback manager to communicate a summary of said feedback to said discussion leader.

8. A computer system for managing an electronically-mediated, asynchronous discussion among knowledge providers, comprising:

- 20 a) a first connective means from a first computer to a second computer;
- b) a second computer;
- c) said first connective means allowing a knowledge provider operating a first computer to capture clip information, encode clip information, and upload an appropriately formatted clip from a first computer to said second computer;
- 25 d) a second connective means allowing said second computer to communicate said appropriately formatted clip at least one viewer; and
- e) a process manager to organize the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested
- 30 by individual viewers and knowledge providers.

9. A computer system as in claim 8, further comprising:

- f) a fourth connective means to allow viewers to communicate feedback to a feedback manager;
- g) said feedback comprising at least one of the group of emails, telephone calls, multimedia and text responses, survey tools, television set-top box technology input; and
- h) said feedback manager communicating said feedback to said discussion leader.

10. A method for managing an electronically-mediated, asynchronous discussion among knowledge providers, comprising the steps of:

- a) providing a first connective means from a first computer to a second computer;
- b) providing a second computer;
- c) providing said first connective means allowing a knowledge provider operating a first computer to capture clip information, encode clip information, and upload an appropriately formatted clip from a first computer to said second computer;
- d) providing a second connective means allowing said second computer to communicate said appropriately formatted clip to a third computer;
- e) providing said third computer; and
- f) providing a third connective means allowing said third computer to communicate said appropriately formatted clip to at least one viewer.

11. A method as in claim 10, further comprising:

- g) providing a knowledge provider serves as a discussion leader and provides additional background information to said knowledge providers through said connective means.

12. A method as in claim 11, further comprising:

- h) providing a process manager to organize the clips into discussion form, authorize viewers and knowledge providers, track which clips have been viewed by individual viewers and knowledge providers and transmit clips as requested by individual viewers and knowledge providers.

13. A method as in claim 11 further comprising:

h) providing a process manager to organize and edit the clips into discussion form, authorize viewers and knowledge providers, track which clips have been
5 viewed by individual viewers and knowledge providers and transmit clips as requested by individual viewers and knowledge providers.

14. A method as in claim 12, further comprising:

- i) providing a fourth connective means to allow viewers to communicate
10 feedback to a feedback manager;
- m) said feedback comprising at least one of the group of emails, telephone calls, multimedia and text responses, survey tools, television set-top box technology input; and
- n) said feedback manager communicating said feedback to said discussion
15 leader.

15. A method as in claim 14, further comprising:

- o) providing summarizing means for sifting, sorting and summarizing said feedback for said feedback manager to communicate a summary of said
20 feedback to said discussion leader.

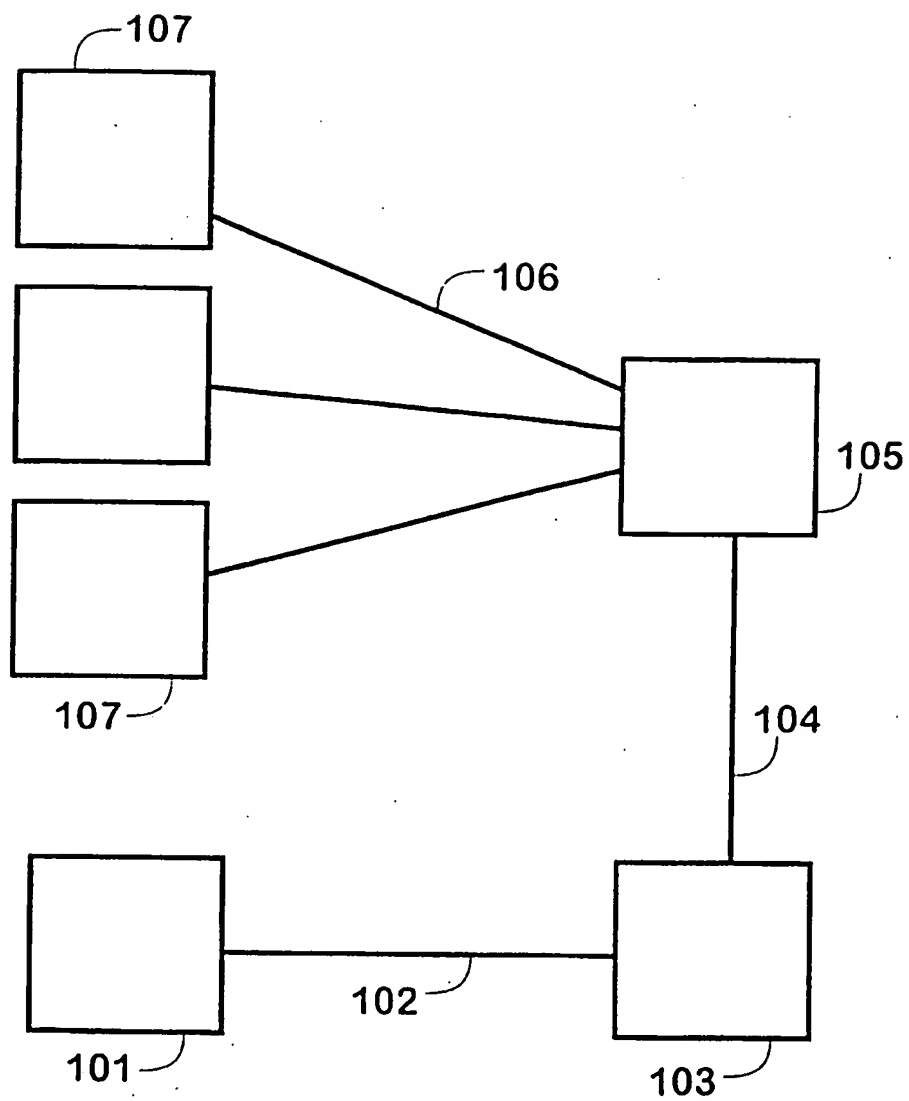


Fig. 1

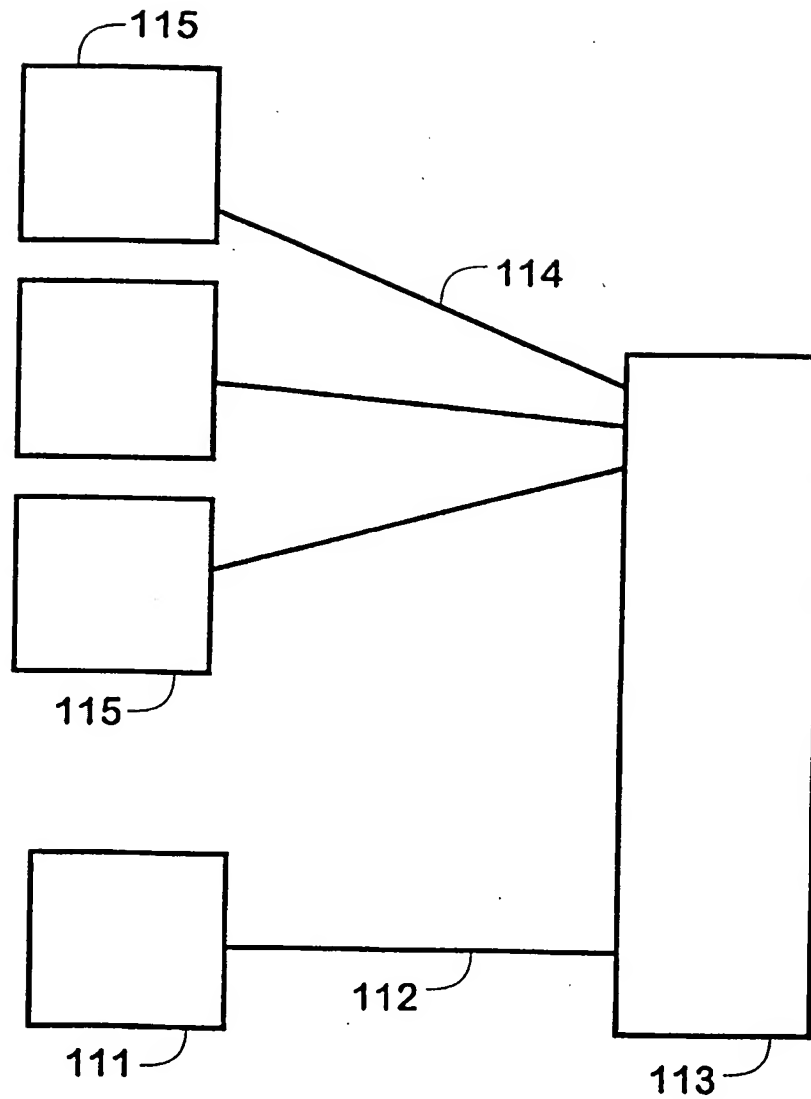


Fig. 1A

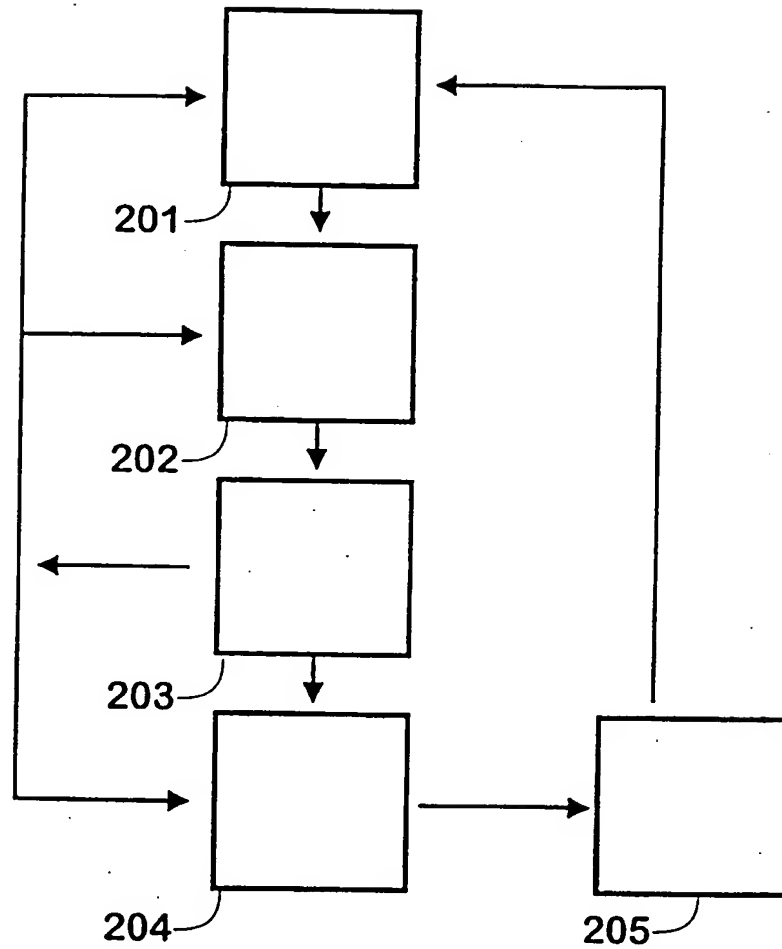


Fig. 2

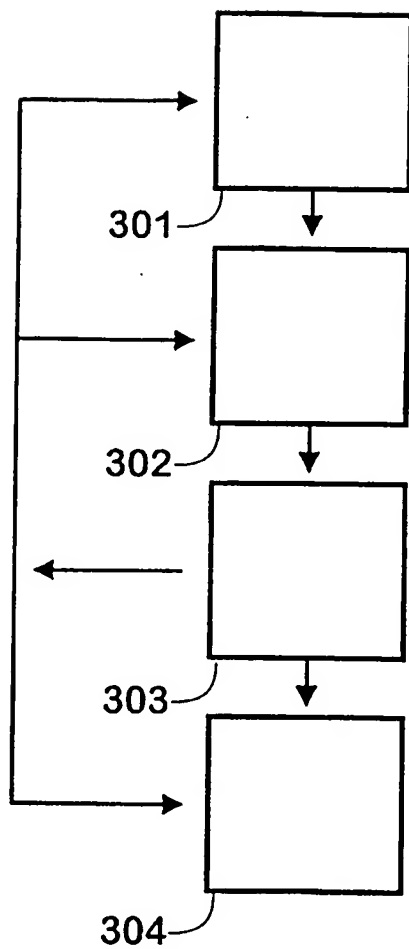


Fig. 3

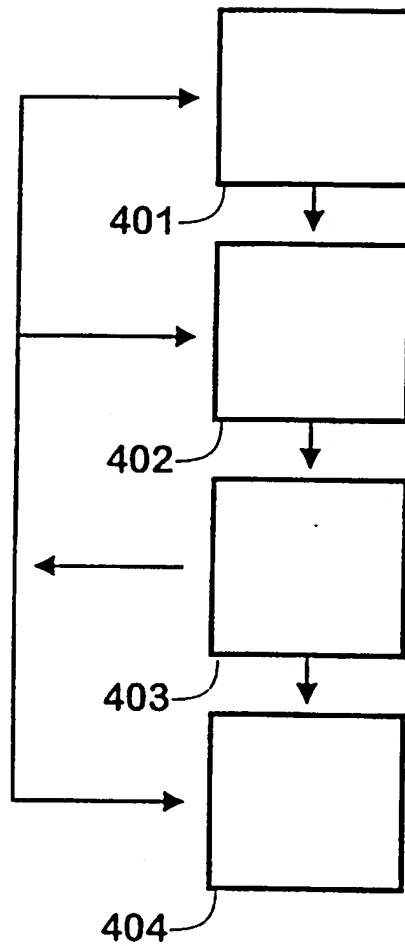


Fig. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/23051

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 3/14, 15/62; G06T 1/00

US CL : 709/217; 348/18; 345/326, 327, 328

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/217; 348/18; 345/326, 327, 328

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,727,156 A (HERR-HOYMAN, et al) 10 March 1998; Abstract; col. 3, lines 7-30; col. 9, lines 1-7; col. 9, lines 32-46	1-15
Y	US 5,675,752 A (SCOTT, et al) 07 October 1997; Abstract; Fig. 2; col. 2, lines 21-62; col. 3, lines 44-58; col. 8, lines 16-60	1-15
Y, E	US 5,966,121 A (HUBBEL, et al) 12 October 1999; Abstract; col. 1, lines 20-60	1 & 10
Y	US 5,706,290 A (SHAW, et al) 06 January 1998; Abstract; col. 2, lines 60-68; col. 3, lines 1-15; col. 4, lines 40-55; col. 20, lines 34-50	3, 6, 8, 9, 10, 12 & 14
Y, E	5, 999,173 A (UBILLOS) 07 December 1999; Abstract	1, 8 & 10

☒ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

19 DECEMBER 1999

Date of mailing of the international search report

02 FEB 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

REUBEN BRO *James R. Matthews*

Telephone No. (703) 305-2399

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/23051

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,109,482 A (BOHRMAN) 28 April 1992; Abstract	1, 8 & 10